Journal of Computational Physics 275 (2014) 154–173

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Cross-code comparisons of mixing during the implosion of dense cylindrical and spherical shells

C.C. Joggerst [a,*], Anthony Nelson [b], Paul Woodward [c], Catherine Lovekin [a], Thomas Masser [a], Chris L. Fryer [a], P. Ramaprabhu [b], Marianne Francois [a], Gabriel Rockefeller [a]

[a] Los Alamos National Laboratory, P.O. Box 1663, MS B227, Los Alamos, NM 87545, United States
[b] University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223-0001, United States
[c] University of Minnesota, 200 Union Street Southeast, Minneapolis, MN 55455, United States

ARTICLE INFO

Article history:
Received 18 January 2013
Received in revised form 30 May 2014
Accepted 11 June 2014
Available online 26 June 2014

Keywords:
Code comparison
Fluid instabilities
Implosion

ABSTRACT

We present simulations of the implosion of a dense shell in two-dimensional (2D) spherical and cylindrical geometry performed with four different compressible, Eulerian codes: RAGE, FLASH, CASTRO, and PPM. We follow the growth of instabilities on the inner face of the dense shell. Three codes employed Cartesian grid geometry, and one (FLASH) employed polar grid geometry. While the codes are similar, they employ different advection algorithms, limiters, adaptive mesh refinement (AMR) schemes, and interface-preservation techniques. We find that the growth rate of the instability is largely insensitive to the choice of grid geometry or other implementation details specific to an individual code, provided the grid resolution is sufficiently fine. Overall, all simulations from different codes compare very well on the fine grids for which we tested them, though they show slight differences in small-scale mixing. Simulations produced by codes that explicitly limit numerical diffusion show a smaller amount of small-scale mixing than codes that do not. This difference is most prominent for low-mode perturbations where little instability finger interaction takes place, and less prominent for high- or multi-mode simulations where a great deal of interaction takes place, though it is still present. We present RAGE and FLASH simulations to quantify the initial perturbation amplitude to wavelength ratio at which metrics of mixing agree across codes, and find that bubble/spike amplitudes are converged for low-mode and high-mode simulations in which the perturbation amplitude is more than 1% and 5% of the wavelength of the perturbation, respectively. Other metrics of small-scale mixing depend on details of multi-fluid advection and do not converge between codes for the resolutions that were accessible.

1. Introduction

In a problem involving the implosion of a sphere or cylinder, Rayleigh–Taylor and Richtmyer–Meshkov instabilities may both be present. The Rayleigh–Taylor (RT) instability develops when the density and pressure gradients in a fluid are op-

* Corresponding author. Tel.: +1 505 665 0528.
E-mail addresses: church@lanl.gov (C.C. Joggerst), anth.nelson@gmail.com (A. Nelson), paul@lsce.umn.edu (P. Woodward), clovekin@lanl.gov (C. Lovekin), tmasser@lanl.gov (T. Masser), fryer@lanl.gov (C.L. Fryer), pramapra@uncc.edu (P. Ramaprabhu), mmfran@lanl.gov (M. Francois), gaber@lanl.gov (G. Rockefeller).

http://dx.doi.org/10.1016/j.jcp.2014.06.037
0021-9991/

posed, as when a denser fluid rests atop a lighter fluid in a gravitational field, or when a lighter fluid accelerates a denser fluid. The Richtmyer–Meshkov (RM) instability develops when a shock passes through an interface between a heavy and a light fluid.

Most simulations and physical experiments to study the details of the growth of these instabilities have been performed in planar geometry [1,2]. However, Rayleigh–Taylor and Richtmyer–Meshkov instabilities also play important roles in non-planar problems. Instability growth on such disparate scales as inertial confinement fusion (ICF) capsules and the interiors of pre-supernova stars may dramatically influence the evolution of these implosions [3,4].

Simulations of instabilities in radially-converging flows present extra challenges beyond those present in simulations of instabilities in planar interfaces; in particular, the ideal spatial coordinate system and grid geometry are no longer obvious. While a spherical grid geometry may seem appealing for following the flow of fluid in spherical implosions, the simplicity of mesh-aligned flow is lost once turbulence develops. Spherical grid geometry also introduces coordinate singularities that impose a preferred orientation on the simulation, and features may develop differently along the axis than at other points around the sphere. In multi-physics problems, such features can expand and contaminate regions far from the axis. Cartesian grid geometries avoid any coordinate singularities and are suitable for simulating a wide variety of physical problems, but using them to simulate spherical implosions raises questions about the spatial resolution required not just for simulating curved interfaces on rectilinear meshes but also for following the growth of perturbations imposed on those interfaces.

Youngs and Williams [5] used a Lagrange-remap code on a 3D spherical polar mesh to simulate turbulent mixing in a sector of a spherical implosion with random perturbations applied initially to the interface between light and dense fluids. The authors found that the width of the mixing zone shrank slightly as they increased the resolution of the mesh but that, at the finest mesh resolution they used, the mixing zone width seemed to have converged.

Do simulations in other coordinate systems, on other grids, achieve similar levels of convergence? To what degree do measurements of the mixing zone width in spherical implosions agree between Cartesian and curvilinear meshes? Do other diagnostics of mixing and turbulence demonstrate the same degree of convergence as mesh resolution increases?

In this paper, we investigate how the choice of grid geometry influences simulations of radially-converging flows by comparing simulations from four codes: RAGE [6], CASTRO [7], FLASH [8–10], and PPM [11]. All four codes model compressible hydrodynamics on Eulerian meshes, but they differ in the details of their discretization schemes, their shock-capturing methods, limiters, and steepeners, their treatment of materials in mixed cells, and their support for different coordinate systems. We assess the impact of the choice of grid geometry on the ability of RAGE and CASTRO to follow the growth of turbulence from perturbed interfaces, and we examine what is required to follow perturbations with different wavelengths and initial amplitudes. We find that perturbation growth is not dependent on the choice of grid geometry, but is influenced by the numerical choices employed in different codes.

Simulations of turbulence in converging flows ultimately require fully three-dimensional spatial meshes [12]. However, comparisons of results between codes can take place using any spatial mesh, and lessons about the advantages or disadvantages of different grid geometries, or the spatial resolution required to reliably measure various diagnostics of mixing and turbulence, should apply to both 2D and 3D simulations. Because 3D turbulence is fundamentally different from 2D mixing, some conclusions drawn from the 2D study may need to be modified for 3D applications. To obtain the most benefit with the least computational cost, we focus in this paper on 2D simulations of cylindrical and spherical implosions.

We examine simulations to which single long-wavelength, single short-wavelength, and multi-mode perturbations were applied. The long-wavelength perturbations give us the largest separation of scales between those we introduce explicitly and those introduced numerically from the mesh, while the short-wavelength perturbations provide us with a way to check the variation of perturbation growth with azimuth as well as a test of the regime in which individual unstable features interact extensively with each other. The multi-mode simulations test both of these regimes simultaneously.

For simulations of cylindrical implosions (where the two-dimensional spatial domain is a slice perpendicular to the axis of the cylinder), we compare results from RAGE and PPM using Cartesian meshes with results from FLASH using a polar mesh. FLASH's polar mesh allows us to simulate an unperturbed converging cylindrical interface; comparing with simulations of unperturbed interfaces on Cartesian meshes exposes the perturbations that arise specifically from the use of a Cartesian mesh, as the curved interface crosses mesh cell boundaries around the cylinder. Once we introduce explicit initial perturbations, the three codes produce similar results; the nonlinear growth of the instabilities ensures that, for sufficiently large initial perturbation amplitude, the imposed perturbations grow faster than any perturbations introduced by the mesh.

For simulations of spherical implosions, we compare results from RAGE, CASTRO, and FLASH using 2D r–z meshes. As in the simulations of cylindrical implosions, we expect there to be an initial perturbation amplitude above which our imposed perturbations grow faster than perturbations arising from the mesh, and we expect that this amplitude will have some dependence on the wavelength of the perturbation and the resolution of the mesh. We vary the amplitude of the perturbation for a fixed problem and fixed maximum resolution to find the amplitude (at a given perturbation wavelength) at which the growth of the imposed perturbation dominates over mesh-induced features.

We describe the initial fluid configuration, the method for driving the implosion, the imposed perturbations, and the mesh resolutions used in the simulations in Section 2. We review the capabilities and algorithmic features of the four codes in Section 3. We compare the results of the simulations, including a variety of diagnostics of mixing and instability growth, in Section 4. We conclude, in Section 5, with a discussion of requirements for simulating perturbed interfaces in radially-converging flows.

Table 1
Initial conditions.

Fluid	Density (g/cm^2)	Specific internal energy (ergs/gm)	C_v
R1	0.05	3.0	2.0
R2	1.00	0.15	0.1
R3	0.10	150	100.0

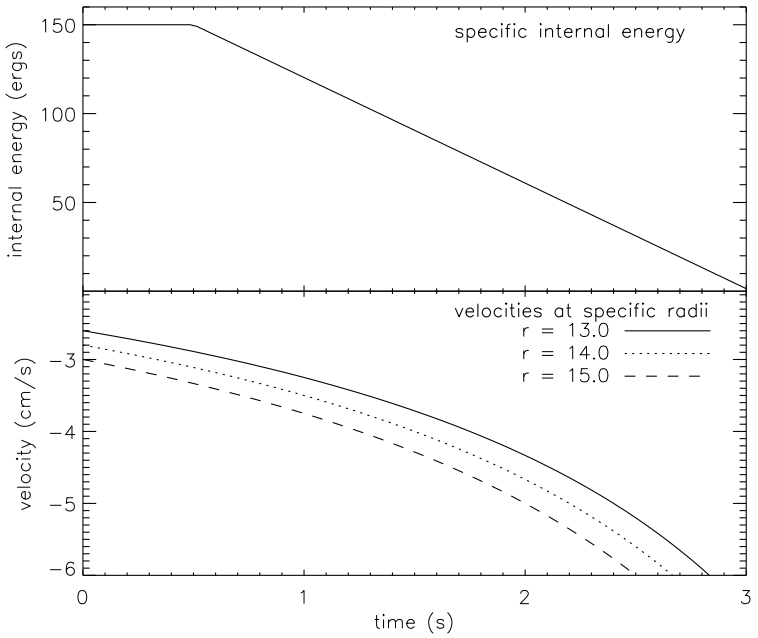

Fig. 1. Conditions imposed in the boundary region as a function of time.

2. Simulating implosions

In this section, we describe our initial configuration, which is the same for all codes. The codes are described in Section 3.

2.1. Initial fluid configuration

The simulations are performed in 2D geometry, and are configured to represent either an imploding cylinder or sphere, depending on whether the simulations were performed on an XY or polar coordinate grid for a cylinder, or an r–z or 2D spherical grid for an imploding sphere. The extent of the computational domain is 15 cm from the origin. The center of the simulation to $R = 10$ cm is initialized as a low-density, low-pressure region. Outside this inner region, from $R = 10$ cm to $R = 12$ cm, is a dense shell with material at the same pressure as the inner region. The implosion is driven by a high-pressure outer material initially located from $R = 12$ to $R = 15$ cm with a density twice that of the inner region. These conditions are summarized in Table 1. All simulations used a gamma-law equation of state with $\gamma = 5/3$.

Because the PPM code uses a moving grid (see Section 3.4) instead of an adaptive mesh refinement (AMR) scheme, the specific internal energy and density as well as the radial velocity are enforced in a "moving boundary region" so that the same boundary condition is enforced in across all codes. The radius beyond which the boundary conditions are enforced changes as a function of time, as described below. The presence of this boundary region ensures that the driving pressure is uniform across codes and grid geometries.

2.2. Driving the implosion—boundary conditions

The density in the boundary region is held constant at the initial value of 0.1 g cm^{-3}. The energy is held constant at 150 ergs for the first 0.5 seconds, then drops linearly to 1.5 ergs (the initial energy in the inner regions) by $t = 3.0$ seconds. The thermal energy in the boundary region as a function of time is plotted in Fig. 1.

The velocity in the boundary region as a function of time and radius is given by

$$V_{bd}(r,t) = -R_o u_{bd}\big(r/R_{bd}(t)\big) \tag{1}$$

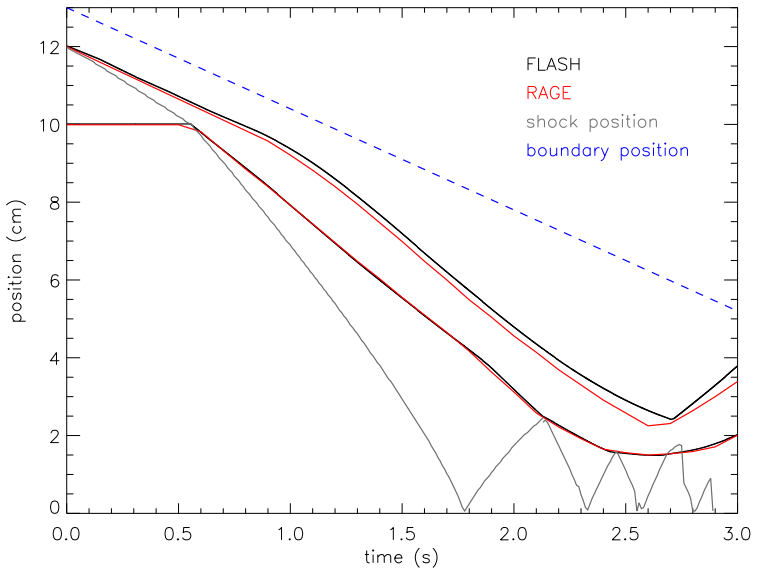

Fig. 2. Cylindrical trajectories for the inner and outer boundaries of the dense shell in 1D in FLASH and RAGE. The position of the shock (in the RAGE calculation) is shown in gray and the position of the boundary is shown as a dotted blue line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where R_o is the initial radius of the boundary, R_{bd} is the boundary position as a function of time, u_{bd} is a constant, r is the radial coordinate, and t is the time. The velocity at three different radii for purposes of illustration is shown in the lower panel of Fig. 1.

The radius beyond which these boundary conditions are enforced as a function of time is given by

$$R_{bd}(t) = R_o(1 - u_{bd}t) \tag{2}$$

where u_{bd} is chosen to be a constant equal to 0.2 s^{-1}.

This is plotted in addition to the 1D trajectory of the dense shell in Fig. 2 and Fig. 3 for the cylindrical and spherical cases, respectively. The boundary region moves inward with time, but never crosses into the dense shell, the computational domain of interest.

2.3. Interface perturbation

Simulations for the cylindrical configuration were performed with four sets of initial perturbations: one in which no perturbations were deliberately introduced (though perturbations arising from the grid were present for codes employing a Cartesian coordinate system), two single-mode perturbations, and a multi-mode perturbation. In the spherical case, two single-mode perturbation calculations were performed. The boundary between the dense shell and the inner fluid was perturbed according to:

$$P(\theta) = A * \cos(m * \theta) \tag{3}$$

where θ is the angle, A is the amplitude (here chosen to be 0.125 cm, or 1% of the wavelength of the longest-wavelength perturbation) and m is the mode number. We investigated a single high-mode order perturbation and a low-mode perturbation in both spherical and cylindrical calculations.

For the case of the cylindrical implosions, modes were chosen such that they were odd and prime, so the final simulation would not have a simple axis of symmetry. We chose $m = 5$ for our low-mode perturbation and $m = 47$ for our high-mode perturbation.

We also initialized one set of cylindrical calculations with a multiple mode perturbation, where modes from $m = 3$ to $m = 33$ were added together. The boundary perturbation was calculated according to:

$$P(\theta) = \sum A_m * \cos\big(m * (\theta + \alpha_m)\big) \tag{4}$$

where the sum is taken over modes from $m = 3$ to 33, A_m is the individual mode amplitude, and α_m is the mode offset. Offsets were chosen randomly, and the individual amplitudes of the perturbations were scaled such that higher-wavelength modes had individual amplitudes that were smaller than the individual amplitudes of the lower-wavelength modes.

For the case of the spherical implosion calculated on a 2D Cartesian or spherical grid, a maximum of 180 degrees of the sphere can be modeled. For these cases we chose even mode numbers for the perturbations similar to those chosen for the cylindrical case above. The low-mode perturbation was perturbed with $m = 6$ and the high-mode perturbation was

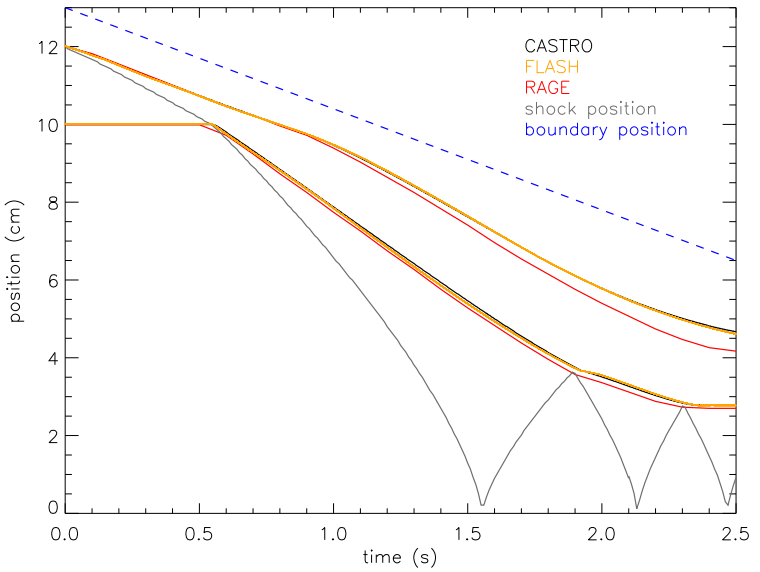

Fig. 3. Spherical trajectories for the inner and outer boundaries of the dense shell in 1D in CASTRO, RAGE, and FLASH. The position of the shock (in the RAGE calculation) is shown in gray, and the position of the boundary region is shown as a dotted blue line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

perturbed with $m = 48$. Since these calculations were performed to study the amplitude at which the imposed perturbation was no longer visible, we did not perform multi-mode calculations for the spherical implosion problem.

2.4. Resolution

Modeling a truly unperturbed interface in a convergent geometry requires the use of a spherical- or polar-coordinate grid. Mapping a spherical or cylindrical problem to a Cartesian grid will introduce small perturbations as the curved interface must be composed of squares cells. The amplitudes and spectrum of these perturbations are proportional to the grid resolution. Increasing the resolution of the simulation reduces the amplitude and shifts the spectrum of these perturbations towards higher mode numbers but does not eliminate the perturbations; as the simulation progresses, these small perturbations will grow and merge through mode competition.

If we were only interested in simulating the behavior of a perfectly unperturbed interface, a one-dimensional (1D) code would be adequate to the task. Real-life problems, however, nearly always present some degree of perturbation.

To capture these perturbations, it is important to ensure that the resolution is high enough that perturbations arising from the grid do not dominate the imposed perturbations. Simulations in the 2D cylindrical implosion case were resolved with 4096×4096 effective resolution. In the Cartesian AMR codes, a base grid of 1024 with 2 additional levels of refinement was chosen; in the case of the polar coordinate simulation, a comparable resolution was employed.

Simulations in the 2D spherical implosion problem were resolved with 2048×4096 effective resolution, with a base grid of 256×512 and 3 additional levels of refinement employed in the Cartesian AMR codes.

3. Codes

All codes are compressible and Eulerian, and all use some version of a higher-order Godunov scheme to advance the hydrodynamic equations, though details of the implementation, the limiters employed, and the formal order vary from code to code. The FLASH code has implemented a 2D polar coordinate computation mode, which we exploit to produce reference polar coordinate simulations. We compare results from FLASH with Cartesian grid codes that use different methods for following the unstable multi-fluid interface on the grid. Mesh refinement in RAGE, FLASH and CASTRO was triggered when the ratio of the second derivative to the first derivative of density or velocity was above a certain threshold value. This discriminator is dimensionless and bounded, and is used in a variety of Eulerian adaptive mesh refinement codes. In RAGE the AMR operates on a cell-by-cell basis; in FLASH and CASTRO groups of cells are flagged for refinement.

3.1. FLASH

FLASH is a publicly available, compressible, Eulerian AMR code. It is a modular code, which is to say, the equation of state, number and characteristics of advected fluids, mesh geometry, and a number of available physics packages can be configured by the user at compile time. It can employ either PARAMESH or CHOMBO to handle block-structured adaptive

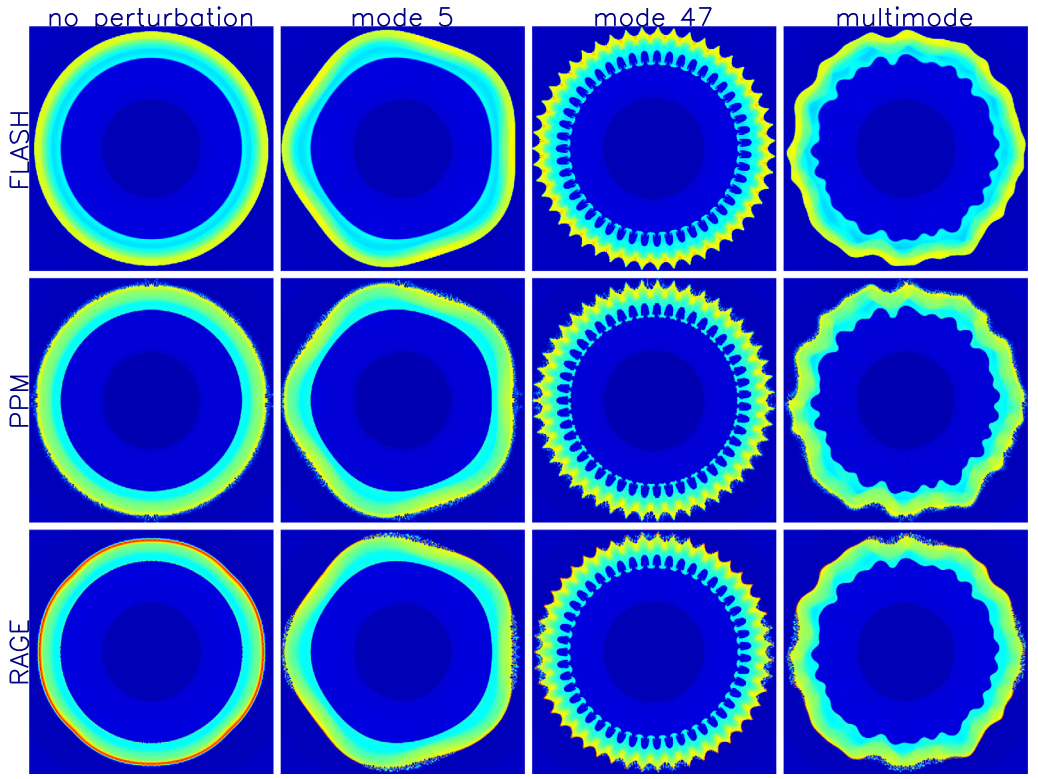

Fig. 4. Snapshots of fluid density in the cylindrical calculations at 1.5 seconds for FLASH (top row), PPM (middle row) and RAGE (bottom row). The inner cylindrical interface is still smooth for the unperturbed calculations in all codes at this point in time. The high-mode perturbation is the only one at this point to experience significant grown, and simulations across codes appear very similar, with only slight variation in the shape of the dense "fingers."

mesh refinement. It can solve the compressible Euler equations using a dimensionally-split PPM scheme, or unsplit WENO or PPM schemes. FLASH can be compiled to solve these equations on a one-, two-, or three-dimensional grid using polar, spherical, or Cartesian coordinates. For these simulations, FLASH was compiled to use a split PPM advection scheme, and to use PARAMESH for adaptive mesh refinement.

FLASH uses PPM-based contact discontinuity steepening to capture shocks. The FLASH code does not explicitly track interfaces between the fluids, so that a small amount of numerical mixing can be expected during the course of the calculation [10].

3.2. CASTRO

CASTRO [7], like FLASH, is a publicly available, modular, compressible, Eulerian AMR code. It uses an unsplit, higher-order Godunov method with characteristic tracing and full corner coupling to compute time-centered edge states to advance the compressible Euler equations. The fluxes of the primitive variables (for CASTRO, these are density, velocity, pressure,energy density, and mass fractions of individual species) are computed using an approximate Riemann solver. The mass fractions are subject to a normalization constraint. The constructions of limited piecewise parabolic profiles of each primitive variable (including abundances) are performed separately for each coordinate direction, using a slope limiting procedure that does not reduce the order of the reconstruction at smooth local extrema. This updated PPM algorithm [13] is insensitive to asymmetries arising from roundoff error. Slopes near shocks are further limited using a flattening coefficient that effectively reduces the Godunov scheme to first-order near strong shocks, which is identical to the approach taken in FLASH. CASTRO employs a small quadratic artificial viscosity to add additional dissipation at strong compressions. This dissipative term is proportional to the divergence of the velocity field. CASTRO's adaptive mesh refinement consists of a nested hierarchy of logically rectangular grids with simultaneous refinement in both space and time.

In 1D CASTRO can be compiled to solve the Euler equations in a spherical or Cartesian coordinate system; in 2D, on a Cartesian r–z or x–y coordinate system, and in 3D, in a Cartesian coordinate system. It employs a modular equation of state and has many available physics packages. The simulations performed for this paper used only the hydrodynamics physics package.

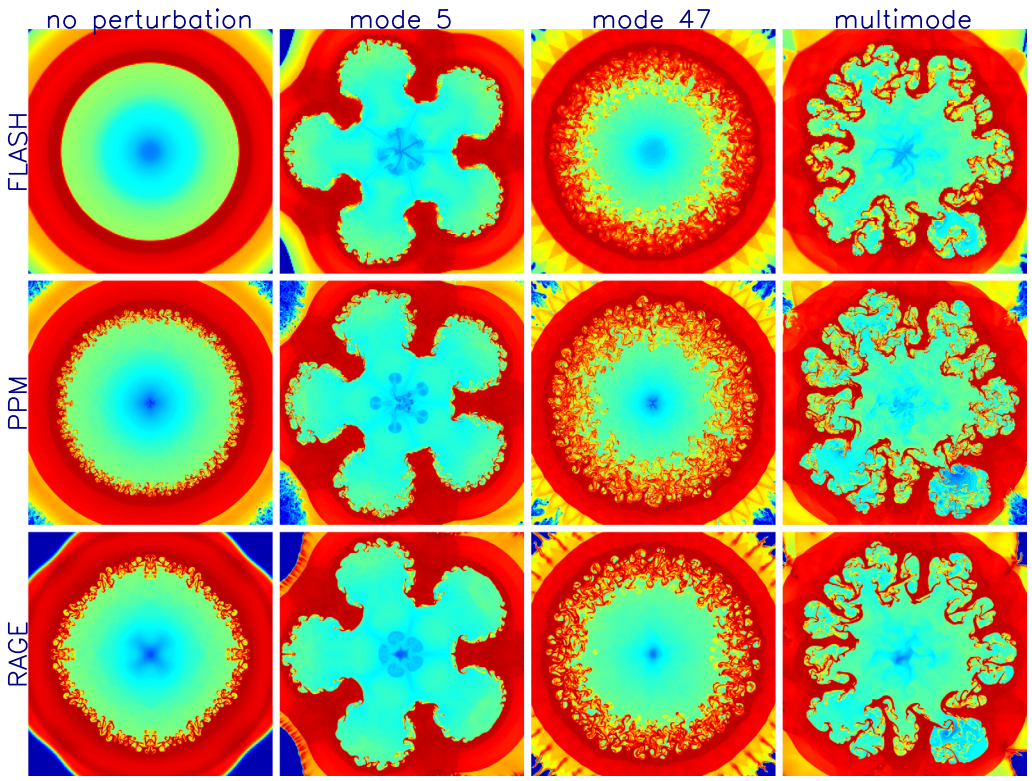

Fig. 5. Snapshots of fluid density in the cylindrical calculations at 2.5 seconds, near maximum compression, for FLASH (top row), PPM (middle row) and RAGE (bottom row). The inner cylindrical interface is preserved for the FLASH simulation (which uses a cylindrical mesh), but small perturbations seeded by the Cartesian grid have grown in the PPM and RAGE calculations, as expected. The simulations for which perturbations were imposed appear similar between codes overall, with differences for a particular perturbation varying more greatly between Cartesian codes (PPM and RAGE) than between the Cartesian codes and the simulations performed on a cylindrical mesh (FLASH).

3.3. *RAGE*

RAGE [6] is a compressible Eulerian AMR code that can be used to simulate one-, two-, or three-dimensional problems. It solves the compressible Euler equations using a dimensionally-split integrator that includes an iterative, HLL-like approximate Riemann solver [14] and multiple slope-limiting options. RAGE employs a tensor artificial viscosity. RAGE, like CASTRO and FLASH, is a multi-fluid code. RAGE employs pressure–temperature–velocity equilibrium in cells that contain multiple materials and offers a variety of interface treatments, most notably an interface preserver [15] and a volume-of-fluids [16] algorithm. For the results presented here, the RAGE interface preserver algorithm was utilized.

3.4. *PPM*

The version of PPM employed in these simulations uses the piecewise parabolic method (PPM) [17,18], a higher-order extension of Godunov's method, to construct interface values between grid cells, and adds a multifluid piecewise parabolic Boltzmann (PPB) advection scheme. The compressible Euler equations are evolved using a directionally split procedure. The version of the PPM scheme that is used in the present code is described in the paper by Woodward [19]. It makes explicit attempts to interpolate characteristic and normal variables in a consistent fashion and not to apply monotonicity constraints when these are not appropriate. A contact discontinuity detection and steepening scheme is included in the method, but this is turned off in the computations shown here, since its function is more accurately carried out by the PPB multifluid advection scheme. PPM offers only 2-D and 3-D Cartesian grid capabilities, but these uniform Cartesian grids can move in order to follow the general radial convergence of the flow studied here.

In PPM, the multiple fluids that may exist in a grid cell are assumed to always have the same pressure and temperature. Along with the gamma-law equation of state, this means that the ratio of the two fluids densities in any given cell is a constant that reflects the ratio of their mean molecular weights. This gives a considerable simplification to the multifluid hydrodynamics scheme. It also permits interpolation of a density variable for a particular fluid that can be continuous and well-behaved across a multifluid interface. The fractional volume of the dense shell fluid is specifically tracked in the PPM code. In each cell the 10 lowest order moments of this variable are specified, and separate partial differential equations, with the form of conservation laws, for each of these moments are explicitly updated by the PPB advection scheme. This

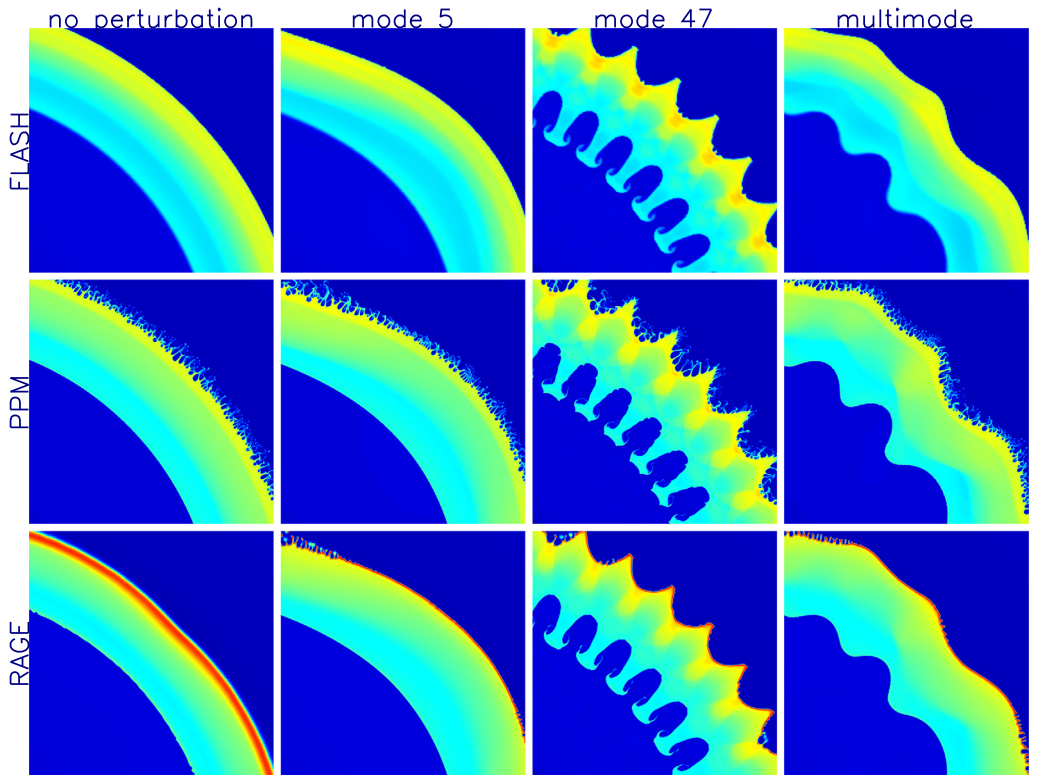

Fig. 6. Detailed snapshots of fluid density in the cylindrical calculations at 1.5 seconds, where the high-mode instability is clearly formed, for FLASH (top row), PPM (middle row) and RAGE (bottom row). The extent of growth of the high-mode instability is similar across simulations performed with different codes, but the shape of the head of the instability varies.

advection scheme has evolved from the original van Leer Scheme VI [20] for unconstrained 1-D advection using parabolas for interpolation of the internal structure of grid cells. This scheme was extended to 2-D in the early 1980s [21], and a constraint added. The present PPB scheme used in our PPM code [21,22] has been extended to 3D and specialized to update only the 10, rather than 27, low-order moments.

The constraints applied in our version of PPB restrict the fractional volume to the range between 0 and 1 but do not apply any other constraints. A choice between two different constraint methods for the interpolation parabola is made that recognizes that when a neighboring cell has its average fractional volume value either 0 or 1, then the value of this variable at the edge of the cell that touches this neighbor should also be 0 or 1, respectively. This enables the PPB scheme to represent sharp interfaces between fluids as very thin, but the representation can never be thinner than a single cell width, because a non-vanishing fractional volume average implies non-vanishing fractional volumes at all points within a cell, except possibly at the edges. The idea is to represent multifluid interfaces as nearly continuous transitions in the fractional volume variable, but to use the extra information and formal accuracy of the PPB scheme to enable this transition to stay comparable to one grid cell in thickness when that is appropriate. There is a necessary trade-off between the thinness of the numerical representation of the transition and the amplitude of small changes in this representation with the phase with respect to the grid cell center. Such small changes can be amplified by a physical instability mechanism, but the unstable modes they introduce are always small in initial amplitude and high in frequency.

The code employs a moving mesh that moves inward at a rate slightly lower than the rate at which the dense shell moves inward. The exterior boundary conditions are thus enforced in a constant part of the grid. The other codes have reproduced the salient features of this moving mesh (i.e. the moving boundary condition) by changing the region in which the boundary conditions are enforced. This boundary is described by Eq. (1). PPM does not employ AMR, so the moving grid allows computational power to be used efficiently as the region of interest shrinks.

4. Results

The codes compared in this paper have different geometric capabilities. RAGE and FLASH can be configured to perform 2D cylindrical and spherical calculations; CASTRO could only be used to perform 2D calculations in spherical geometry, not cylindrical geometry, at the time the calculations were performed and PPM can be used in 2D to perform cylindrical calculations but not spherical calculations. All codes can perform calculations in 3D Cartesian coordinates. Since this study was

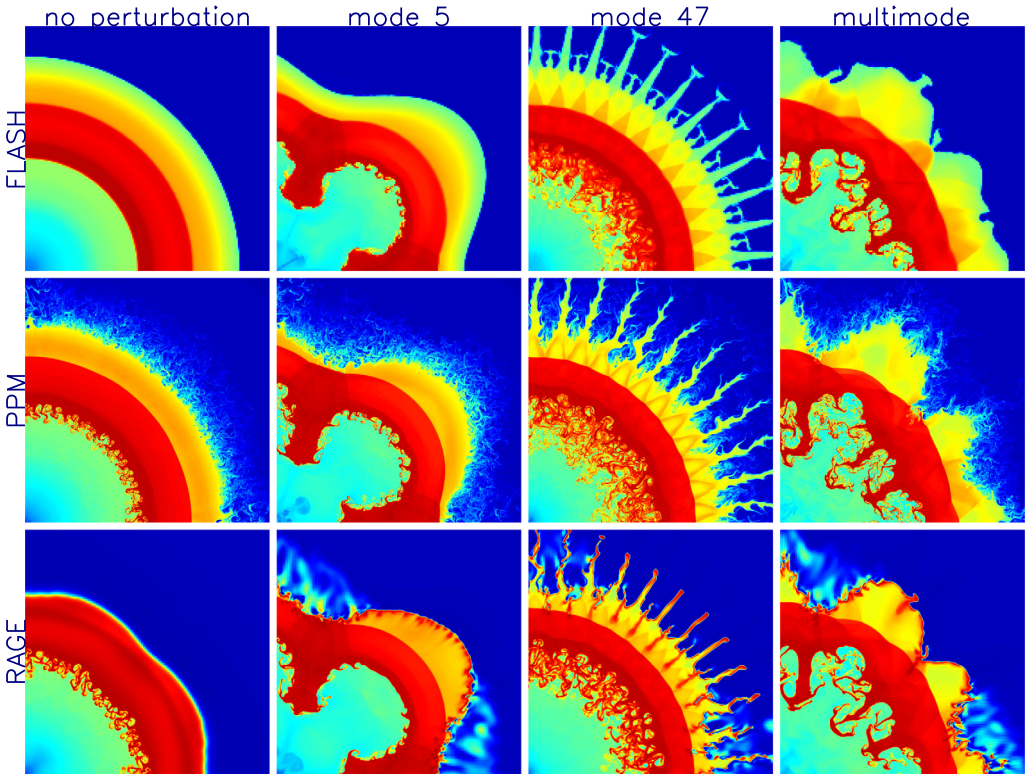

Fig. 7. Detailed snapshots of fluid density in the cylindrical calculations at 2.5 seconds, near maximum compression, for FLASH (top row), PPM (middle row) and RAGE (bottom row). The overall extent of fluid growth appears similar across simulations performed with different codes, but the PPM simulations show more mixing (represented by yellow regions) and secondary instability growth tilted toward higher modes than the RAGE simulations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

primarily intended to assess the potential pitfalls and advantages of performing calculations similar to the ones presented in this paper in 3D, we compare codes across both cylindrical and spherical 2D geometries.

4.1. Cylindrical geometry

Fig. 4 shows a series of images of fluid density for the cylindrical geometry simulations at 1.5 seconds of simulation time, during the early stages of the growth of the high-mode instability but before the low-mode instability has grown substantially. Fig. 5 shows a series of fluid density images at 2.5 seconds of simulation time, near maximum compression for all cylindrical geometry simulations. The densities have been scaled to best show the growth of perturbations; the late time densities are about 13 times higher than the initial density. The unperturbed case, shown in the first column, shows the greatest difference between the Cartesian- and cylindrical-mesh simulations. The smooth interface is preserved in the cylindrical-mesh FLASH simulation, while instability growth is present in the PPM and RAGE calculations. This is an expected effect. Small perturbations arising from the Cartesian grid have grown and merged, resulting in instability growth by the point of maximum compression in the simulation.

This RAGE calculation also shows the most obvious numerical artifact of any calculation in this paper. The bottom row of images in Fig. 5 show a distinct distortion of the dense shell, most especially in the unperturbed calculation and the high-mode calculation. It is not as apparent in the multi-mode or low-mode calculation. This is not a product of choice of grid geometry, as the other Cartesian code, PPM shows no such strong effect; it is most likely a result of the combination of weak artificial viscosity and boundary-condition matching in RAGE.

The overall shape and degree of growth of the RT (and RM) instabilities in the perturbed simulations appear very similar between the results computed in cylindrical geometry (first row) and the Cartesian simulations (bottom two rows). This similarity is most easily seen for the high-mode simulations in Fig. 4, and for the low-mode and multi-mode simulations in Fig. 5. This implies that for a large-enough amplitude initial perturbation and sufficient resolution, the simulation results will be the same in both polar and Cartesian code geometries.

Figs. 6 and 7 show details of the instability growth and development at 1.5 and 2.5 seconds, respectively. In Fig. 6, the greater initial amplitude relative to wavelength of the high-mode perturbation has ensured that this perturbation has grown considerably. The RT “fingers” are well-developed, but the shape of the ends shows variation from code to code, resulting from algorithmic differences among the codes. Otherwise identical simulations performed with PPM at different resolutions

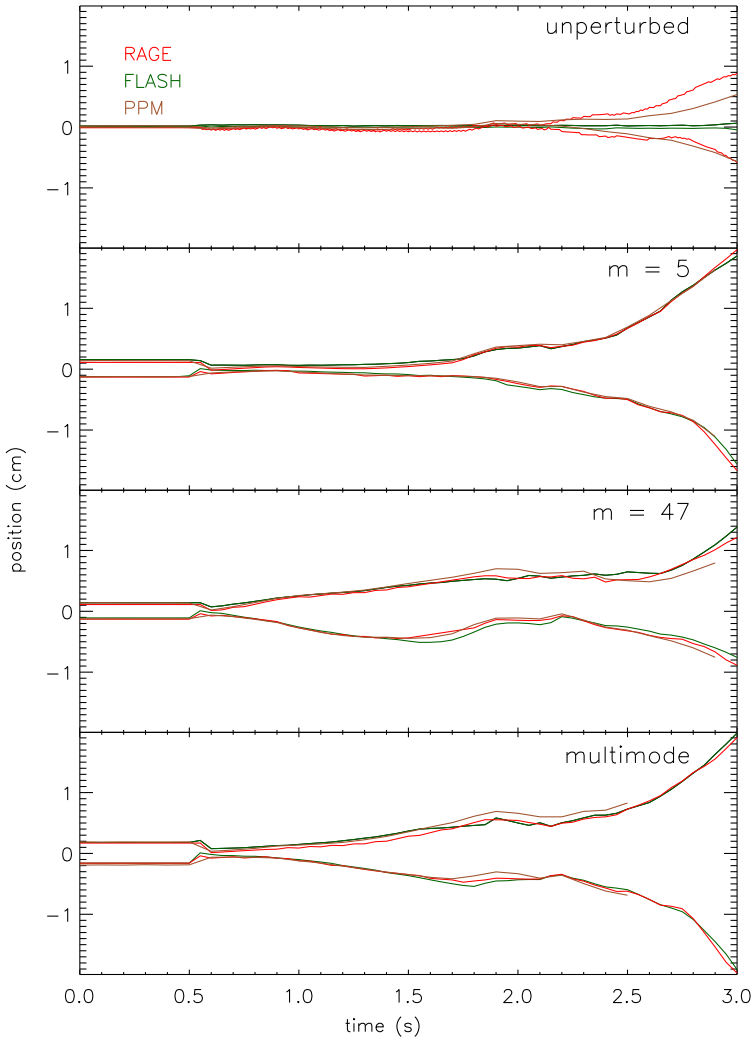

Fig. 8. Bubble/spike amplitudes (distance from 1D trajectory) for cylindrical calculations as a function of time for all codes for four different perturbations. The top panel shows the unperturbed case, in which the Cartesian codes (RAGE, FLASH, and PPM) show perturbation growth, but the code employing polar coordinates (FLASH) does not, as expected. For the low-mode (second panel from top), high-mode (third panel from top) and multiple mode (bottom panel) perturbations, agreement between Cartesian codes and the polar-coordinate codes is excellent, implying that choice of grid geometry does not have a large effect on the overall amplitude of perturbation growth.

also show changes in finger shape with resolution, so refinement schemes and resolution may play a role, as well. A similar effect has been documented in the paper by Almgren [7] for CASTRO when different limiters and advection schemes are employed. By a simulation time of 2.5 seconds, as shown in Fig. 7, the low-mode perturbations have grown considerably, and the high-mode perturbations have begun to interact. Differences between the calculations performed by various codes are now visible in the apparent mode-order of the secondary instabilities in the low-mode case. The secondary instability growth for the low-mode PPM calculations appear to be tilted toward higher mode numbers than those in the analogous RAGE calculations. More small-scale mixing appears to have occurred in the PPM and FLASH calculations than in the RAGE calculations, which is likely due to the explicit attempts to limit numerical diffusion in the RAGE code. Though the appearance of the outer boundary also differs between codes, it does not appear to influence the behavior of the inner boundary between the dense shell and the light inner fluid, which is the focus of this study.

Fig. 2 shows 1D cylindrical implosion trajectories calculated by RAGE and FLASH. These 1D results provide a trajectory against which the bubble and spike amplitudes will be measured. They also allow us to determine the 1D Atwood number as a function of time, where the time-dependent 1D Atwood number is found by assuming a constant density in the inner region and dense shell, and using the relative position of the inner and outer shell boundaries to determine these densities.

A point of interest in designing ICF capsules as well as assessing supernova dynamics is the degree of growth of the RT and RM instabilities. This is most often measured by assessing the heights of the “bubbles” and “spikes” of instability growth. Here, we define the bubble height as the maximum radius where the average concentration of the light fluid is greater than 1%, and the spike height as the minimum radius where the average concentration of the dense fluid is greater

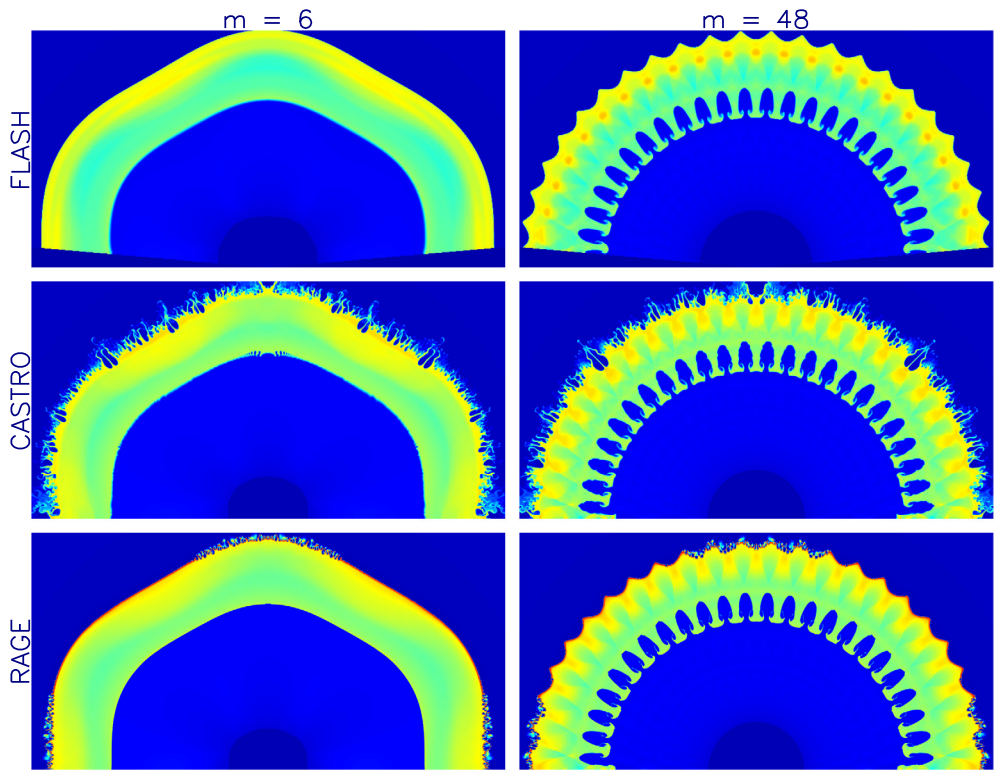

Fig. 9. Snapshots of fluid density in the spherical simulations at 1.5 seconds, before maximum compression, for FLASH (top row), CASTRO (middle row) and RAGE (bottom row). The high-mode instability is already well-formed at this time. While the shape of the high-mode instability differs slightly between codes, as in the cylindrical case, such shape differences are a result of advection scheme differences, rather grid geometry.

than 1%. Fig. 8 shows the positions of the bubble and spike fronts relative to the 1D position of the inner interface for all codes as a function of imposed perturbation. The top panel shows the results for the unperturbed case, showing that bubble and spike height are essentially zero for the calculation performed on the curvilinear grid, as expected. The bubble and spike positions for the Cartesian codes are comparable for the unperturbed case, but they are non-negligible.

For the perturbed cases, however, the bubble and spike trajectories are quite similar, with the differences between Cartesian codes being similar to differences between FLASH and the Cartesian codes. Grid geometry seems to have no influence on the bubble and spike amplitudes for these perturbations. All codes have converged to the same mixing-zone width in the mode 5 case, where the imposed amplitude was 1% of the perturbation wavelength and 33 times the smallest grid cell size. The codes have achieved good agreement in the mode 5 case. Agreement in the mode 47 case is not as good as that in the mode 5 case, in part because at higher mode numbers the perturbation wavelength is resolved with fewer grid cells. Also, in convergent geometry, interactions between the individual perturbations become important and more resolution is generally required. The multi-mode case falls somewhere between the high- and low-mode cases.

4.2. Spherical geometry

2D spherical geometry differs from 2D cylindrical geometry in a significant way. In 2D cylindrical geometry, all the cells with the same surface area have the same volume—the depth is presumed to be the same. This is not true in 2D axisymmetric/spherical geometry. In this geometry, the volume of a cell depends on its x coordinate, as all cells are defined to be tori rotated about the y axis. A cell at a small x coordinate (near the y axis) has a smaller volume, and thus comprises less fluid, than a cell far from the y axis, at high x coordinate. The volume of a cell is given by

$$v \equiv 2\pi x dx dy \tag{5}$$

where v is the volume, x is the x-coordinate, and dx and dy are the x and y extents of the cell, respectively. This means that nearly all axially symmetric codes, in polar or Cartesian coordinates, will show some degree of "flow" along the y axis, since there is simply much less volume in that region. This geometry also means that a maximum of 180 degrees can be modeled, and that a reflecting boundary condition must be imposed on the Y-axis.

Because only 180 degrees of the sphere could be modeled, we imposed even perturbations ($m = 6$ and $m = 48$) on the inner surface of the dense region so that the simulations would be symmetric across the equator. Fig. 9 shows the appearance of the instability after 1.5 seconds of simulation time in all codes and for both perturbations, and Fig. 10 shows

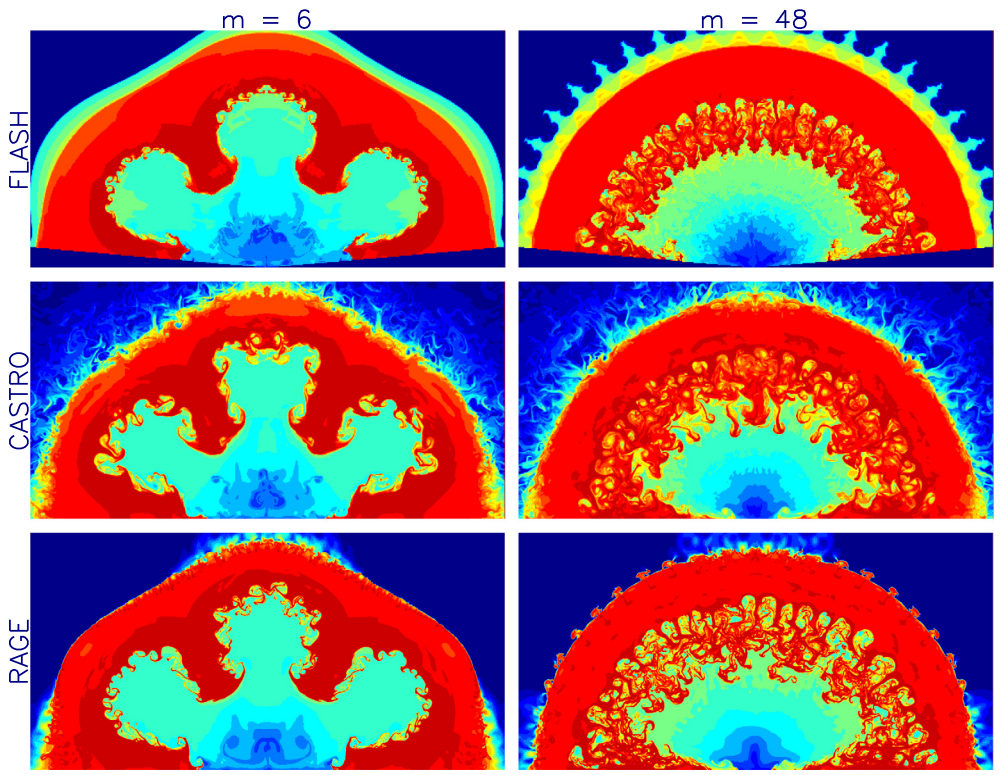

Fig. 10. Snapshots of fluid density at 2.5 seconds in the spherical simulations, after maximum compression, for FLASH (top row), CASTRO (middle row) and RAGE (bottom row). The growth of higher-mode instabilities in the low-mode perturbation simulations (left panel) differs between the Cartesian codes RAGE and CASTRO, with CASTRO calculations showing more dramatic growth of secondary perturbations and a spectrum tilted toward lower mode numbers than RAGE.

the appearance of the instability at 2.5 seconds of simulation time. The shape of the instability appears similar between codes for both perturbations. Again, the same scaling is applied to both images; late-time densities are about 7 times greater than the initial density. A 5 degree cone about the positive and negative y-axis was omitted from the FLASH calculations, which is why the dense shell does not extend to the bottom of the plot. The regions were omitted because the small volumes at the poles posed difficulties for the code. At early times (Fig. 9) the instability in the high-mode perturbation has grown significantly, though the low-mode perturbation has yet to grow. At later times (Fig. 10) low-mode perturbation growth can be compared. Mixing on small scales can be traced by looking at the yellow regions of the plot. It is apparent that more mixing on small scales takes place in CASTRO than in RAGE, which is a consequence of algorithmic details rather than mesh geometry. On the whole, one cannot tell by looking at the instability which was computed with a Cartesian and which with a spherical grid geometry.

Fig. 10 at 2.5 seconds shows more discrepancies between the codes than does Fig. 9. The low-mode perturbation has clearly grown at this point, and there is more growth of secondary instabilities along the main perturbation in the Cartesian codes than in the spherical-coordinate code. The spectrum of these perturbations appears different between CASTRO and RAGE. The spectrum of secondary perturbations in the CASTRO simulations, as in the PPM simulations, is tilted toward higher mode numbers than is the spectrum in the RAGE simulations. CASTRO also shows more mixing on grid-cell scales. Details of the instabilities at 1.5 and 2.5 seconds are shown in Figs. 11 and 12, respectively. As for the high-mode calculations, the shape of the high-mode fingers is different between codes, as is apparent in Fig. 11 at 1.5 seconds. By 2.5 seconds, the high-mode perturbations have begun to interact and the low-mode perturbations have grown significantly. The RAGE code, which employs an interface preservation scheme to limit numerical diffusion, shows less mixing on small scales than do simulations from codes with no such scheme. On the whole, however, the bubble and spike amplitudes and appearance of the instability appear similar across codes.

Fig. 13 shows the locations of the average bubble and spike positions relative to the 1D trajectory of the inner surface of the dense shell (shown in Fig. 3) as a function of time. The upper panel shows the evolution of the low-mode ($m = 6$) perturbation, while the bottom panel shows the evolution of the high-mode ($m = 48$) perturbation. Both the low- and high-mode perturbation simulations agree quite well between codes. Agreement between the codes at late times is not as good as the agreement between codes at late times in the cylindrical calculations; this is likely due to numerical artifacts originating at the boundary along the Y-axis of the simulation impinging on the rest of the simulation. The flow along the Y-axis is visible in all codes (see Figs. 9 and 10) and becomes more prominent as the simulation progresses. The numerical

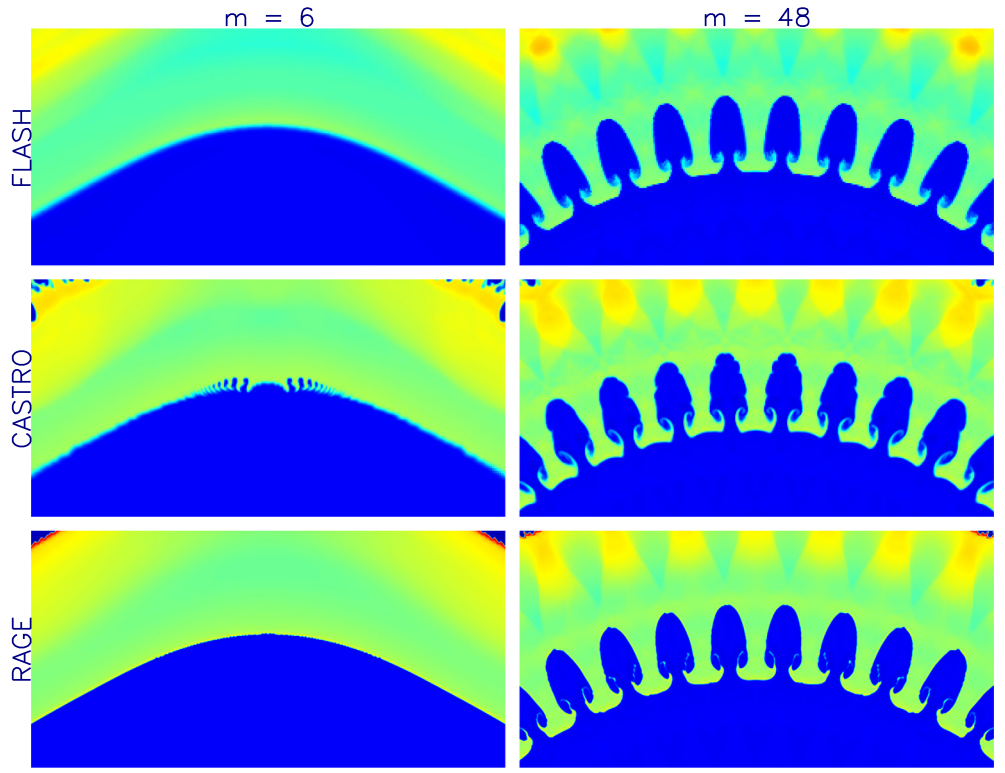

Fig. 11. Detailed snapshots of fluid density at 1.5 seconds, before maximum compression in the spherical simulations, for FLASH (top row), CASTRO (middle row) and RAGE (bottom row). The extent of growth of the high-mode perturbation is similar across calculations performed with different codes, but the shape of the dense "fingers" varies slightly. The FLASH mesh excludes a 5-degree cone around each pole.

artifacts introduced by this boundary are at least partially responsible for the larger variation in bubble-spike amplitude in spherical as opposed to cylindrical geometry.

4.3. Minimum resolution and perturbation amplitude

Previous sections showed that all codes converge to a limit solution with regard to the overall height of the instability with the exception of the unperturbed case, as expected. We have further investigated the resolution needed to resolve a given perturbation amplitude for a given wavelength of perturbation with RAGE and CASTRO. Such a comparison using these two codes is especially interesting because, while both employ Godunov-based schemes on a Cartesian grid, RAGE employs an interface preservation scheme to further reduce numerical dissipation on material boundaries, while CASTRO makes no such provision.

The minimum amplitude at which a particular perturbation will dominate perturbations introduced by an unaligned grid is of interest. One would expect this minimum amplitude to vary with both perturbation wavelength and grid resolution. More finely resolved grids should introduce smaller perturbations, and hence we would expect smaller minimum amplitudes on more finely resolved grids.

We have performed calculations with 1, 2, and 3 refinement levels on a 500×1000 base grid for the both the high- and low-mode perturbations in r–z coordinates with RAGE. We have performed calculations with 3 refinement levels on a 512×1024 base grid for both the high- and low-mode perturbations in r–z coordinates with CASTRO.

For this particular problem, instability growth is characterized by two phases: early, linear growth, and late-time nonlinear development. The early phase is relatively simple, and can be effectively characterized at a very basic level by whether or not the imposed perturbation dominates instability growth.

The time at which the linear growth phase is supplanted by a nonlinear phase will depend on both perturbation amplitude and initial perturbation wavelengths.

Fig. 14 shows the normalized number of instabilities observed as a function of resolution and wavelength for the mode 48 perturbation N_f, where N_f is defined to be

$$N_f \equiv \frac{N_d - N_e}{N_e} \tag{6}$$

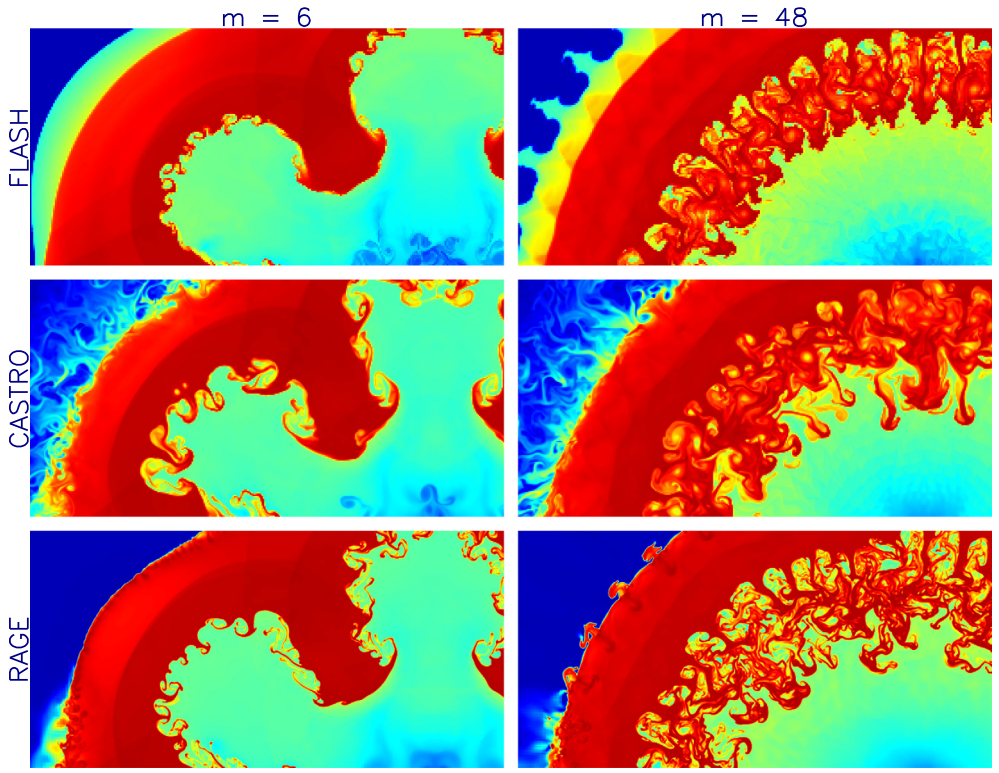

Fig. 12. Detailed snapshots of fluid density at 2.5 seconds, after maximum compression in the spherical simulations, for FLASH (top row), CASTRO (middle row) and RAGE (bottom row). RAGE shows less secondary instability growth in the low-mode case than CASTRO, and shows less diffusion of high-density material than CASTRO, as well.

where N_d is the number of instability "fingers" seen in the simulation that do not show departures from symmetry and N_e is the number of instability fingers one would expect based on the imposed perturbation. For a 180 degree r–z simulation with an imposed mode 48 perturbation, this number is 24.

Ideally, at early times, a code would show the same number of symmetric instability fingers as were put in the initial perturbation. But our codes are not perfect, and even the spherical-coordinate codes don't have meshes perfectly aligned with the perturbed interface. Small, high-mode perturbations will be introduced by any given grid to a perturbed interface. Fewer asymmetric "fingers" are observed at lower resolution than higher resolution for a given perturbation amplitude because the less-resolved simulations have a higher effective numerical viscosity than the more resolved simulations, which serves to damp out small secondary perturbations. Also, the more resolved grids introduce higher-mode-number noise into the simulation than less-resolved grids.

In addition, the number of asymmetric fingers is not constant from code to code. Rather, it is governed by algorithmic details that differ between codes. RAGE's interface preserver has the effect of suppressing both diffusion and the growth of small perturbations, which can be seen in the number of affected fingers being generally lower, and thus N_f tending toward zero, than is observed in CASTRO. Assuming a small amount of distortion in less than 10% of the "fingers" is acceptable, the highest resolution simulations for both CASTRO and RAGE are sufficiently resolved at perturbation amplitudes 7 times the size of the smallest grid cell, or 2% of the perturbation wavelength.

These secondary fingers are still present in a spherical coordinate grid, since even a spherical grid is not aligned with the perturbed interface. This can be seen in Fig. 5, where small additional instabilities for the mode 6 perturbation are visible in both the Cartesian and spherical mesh simulations. When the wavelength of the perturbation is sufficiently long in relation to the wavelength of the perturbations arising from the grid, the high-frequency grid perturbations will not be evident in the final results.

Once the perturbation has grown slightly, it can also be effectively characterized by the bubble and spike amplitudes of the instability. At this point in the simulation, relatively little small-scale mixing has taken place; in other words, regions that are composed of roughly equal parts of both heavy and light fluids comprise a (vanishingly) small part of the computational domain.

We characterize the differences between simulations with different initial amplitudes and resolutions at the linear growth phase in Fig. 15. Bubble-to-spike distances from both CASTRO and RAGE are shown at times corresponding to near the end of the linear growth phase. At the highest perturbation amplitudes, it can be seen that the codes are in good agreement. For the low-mode case, the codes agree for perturbation amplitudes of 0.125 cm (1% of the wavelength of the

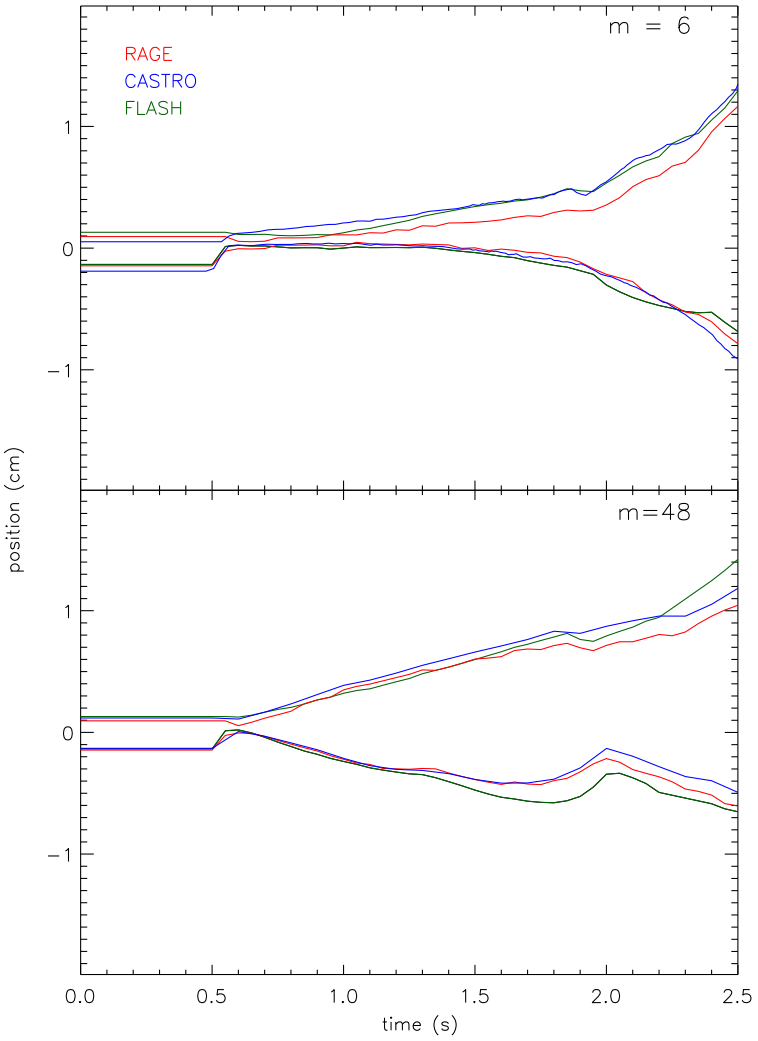

Fig. 13. Bubble/spike amplitudes (distance from 1D trajectory) for the spherical simulations as a function of time for all codes for low- and high-mode perturbations. The codes show good agreement, though RAGE calculations have an upper trajectory slightly interior to that of the other codes, as in the 1D case. The choice of grid geometry has a minor impact on the overall amplitude of instability growth for both low- and high-mode perturbations.

initial perturbation) and all resolutions. At perturbation amplitudes below 1%, the growth of perturbations arising from the grid dominates the growth of the imposed perturbation; the codes tend to plateau to a bubble-spike amplitude that is unique to the code and the simulation time. This plateau amplitude is dependent on the internal details of the code. CASTRO plateaus higher than RAGE, since the former exhibits a greater degree of numerical diffusion which begins to dominate the bubble/spike amplitude. For the high-mode case, the codes agree for perturbation amplitudes about 0.05 cm (5% of the wavelength of the initial perturbation). Both codes plateau at lower absolute perturbation amplitudes for the low-mode perturbation, since above a certain resolution the amplitude to wavelength ratio determines the degree to which the perturbation will grow.

The latter phase of instability development is by comparison more complicated. At this point, the instability has interacted with multiple reverberating shocks. Because the region of interest is converging, the individual “fingers” of the instability have drawn closer together and may have begun to interact, depending on the initial wavelength of the perturbation. Bubble and spike amplitudes no longer capture the salient details of the calculation. The resolved nature of the calculations at late times is perhaps best characterized by parameters that quantify the “mixedness” of the simulation within a particular region.

These simulations indicate that bubble/spike amplitudes will converge where the initial perturbation amplitude is at least 1.0% of the wavelength (0.125 cm) for low-mode perturbations. For the high-mode case, where the wavelength of the perturbation is comprised of a factor of 10 fewer grid cells, bubble-spike amplitudes will converge when the initial perturbation amplitude is at least 5% of the perturbation wavelength (0.05 cm). For realistic astrophysical cases, we expect perturbations on the order of 5–10% of the wavelength of the perturbation, well above this limit.

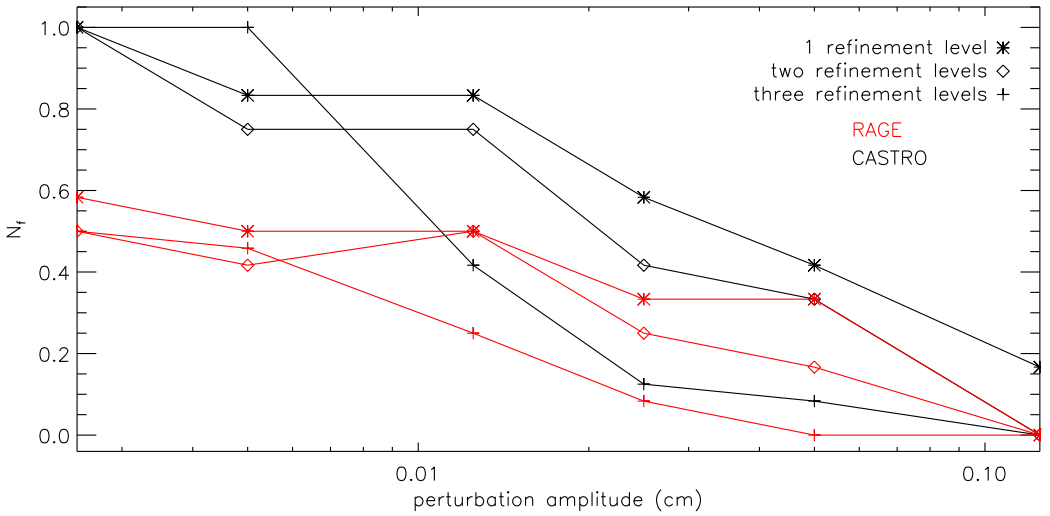

Fig. 14. Normalized number of instabilities error N_f (see Eq. (6) for definition) observed at a given resolution and a given amplitude for the mode 48 perturbation in 2D r–z coordinates. RAGE results are shown in red and CASTRO results are shown in black. In RAGE, a smaller number of distorted fingers is observed at both higher amplitude and higher resolution for amplitudes larger than 0.05, or 5% of the wavelength of the perturbation. The number of asymmetric instabilities observed below this threshold is dominated by noise from the grid. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The advection schemes employed by different codes will also affect the resolution and perturbation amplitude at which simulations appear converged (if at all) in metrics other than the bubble/spike amplitude. The influence of these numerical schemes is shown in Fig. 16. Fig. 16 shows the total mass of light fluid binned by the mass fraction of light fluid present in a given cell for mode 6 (left) and mode 48 (right) simulations at times from the onset of mixing through the end of the calculation. The apparent integral looks smaller for RAGE than for CASTRO, especially in the low-mode simulation, since much of the mass was unmixed and thus appears as a vertical line at the right of the plot. RAGE shows much less mass in the mixed bins nearer the center of the figure than does CASTRO at nearly all times during the simulation. This is a direct result of RAGE's interface preservation scheme and the fact that it advects individual materials separately, while mass and a mass abundance vector are advected in CASTRO. This leads to less numerical diffusion in the former than the latter.

At later times, the interaction of the instabilities with each other and with the multiple shocks in the simulations becomes significant as a driver of mixing. This is reflected in the agreement between the CASTRO and RAGE results for the rightmost bins by 2.2 seconds for mode 48 and 2.5 seconds for mode 6; mixing occurs earlier in mode 48 than mode 6, because of the greater perturbation amplitude relative to wavelength in the former than in the latter. CASTRO shows a large peak in the mass of light fluid present in bins dominated by heavy fluid, which is not as prominent in the RAGE simulations. This peak especially pronounced at late times in the mode 48 simulation, and is a result of the greater amount of numerical diffusion in CASTRO than in RAGE.

Fig. 17 shows two different parametrizations of the "mixedness" of a given simulation at a given time as a function of amplitude. The top panels show Q_a, which is defined to be

$$Q_a \equiv \frac{\sum_{0.05<F_h<0.95} M_h}{\sum_{0.05<F_h<0.95} M} \tag{7}$$

where F_h is the mass fraction of the cell composed of heavy fluid, M_h is the mass of the heavy fluid, and M is the total mass in the cell. This quantity gives an effective abundance of the heavy fluid in regions that are mixed at greater than 5%. The bottom panel shows Q_v, which is defined to be

$$Q_v \equiv \sum_{F_h<0.05} V \tag{8}$$

where V is the volume of a given cell. Q_v gives a measure of the volume occupied by unmixed or only very slightly contaminated light fluid. A greater degree of convergence would be indicated by the successively higher refinement levels being closer to one another.

The quantities shown in Fig. 17 are dependent to a greater or lesser extent on details of how the individual codes handle fluid interfaces and advection. CASTRO shows a greater degree of diffusion or mixing on small scales, which can be seen directly in Figs. 10 and 12 and in histogram form in Fig. 16. This is apparent in the top panel of Fig. 17, where CASTRO shows a slightly smaller value for Q_a than RAGE in the mode 48 calculations, but a significantly smaller value in the mode 6 calculations, where less self-interaction-driven mixing has taken place and diffusion has played a greater role.

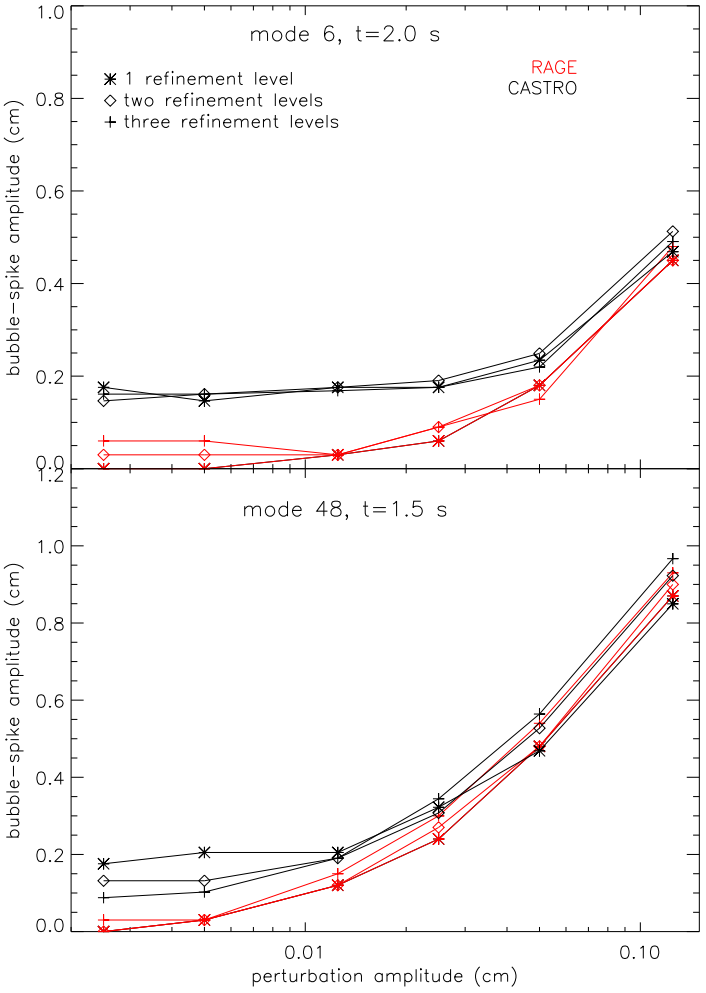

Fig. 15. Bubble-to-spike distance during the linear growth phase for mode 6 (top) and mode 48 (bottom) perturbations in both RAGE (black) and CASTRO (red). These simulations indicate that bubble/spike amplitudes have converged where the initial perturbation amplitude is at least 1.0% of the wavelength (0.125 cm) for low-mode perturbations. For the high-mode case, where the wavelength of the perturbation is resolved by a factor of 10 fewer grid cells, bubble-spike amplitudes will converge when the initial perturbation amplitude is at least 5% of the perturbation wavelength (0.05 cm). For perturbation amplitudes below these values, the perturbation seen in the simulation is dominated by grid noise and so plateaus to some value for a given code and resolution. RAGE results plateau at lower values than do the CASTRO results at perturbation amplitudes low enough that perturbations arising from the grid dominate, an effect most likely caused by the lower numerical diffusion in RAGE. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The bottom panels of Fig. 17 show the volume of fluid that is "uncontaminated" below 5%, and is even more sensitive to details of fluid advection than the first panel. The greater amount of diffusion in CASTRO is apparent here, in a much smaller uncontaminated value at all amplitudes for the mode 6 perturbation in RAGE. The mode 48 perturbation shows that at low perturbation amplitudes, where the instabilities are dominated by grid noise, RAGE shows a higher unmixed volume than CASTRO. The codes converge to a similar value for this metric at high perturbation amplitudes, when the mixing is dominated by the instability fingers interacting with one another.

For problems involving (thermonuclear) combustion, such as ICF capsules and supernovae, metrics which quantify the mixing at small scales are potentially quite important. The values of quantities like Q_v and Q_a are insensitive to spatial resolution but highly dependent upon differences in multifluid advection algorithms.

5. Conclusion

We have presented 2D results from four different codes comparing the evolution of RT and RM instabilities arising from perturbations in cylindrical and spherical implosions. In broad terms, agreement between spherical/cylindrical and Cartesian grid geometries is quite good. Perturbation growth is more influenced by the details of the advection scheme in a particular code than by grid geometry. The FLASH code, which employed spherical geometry, gives results which are consistent with results from the other Cartesian codes.

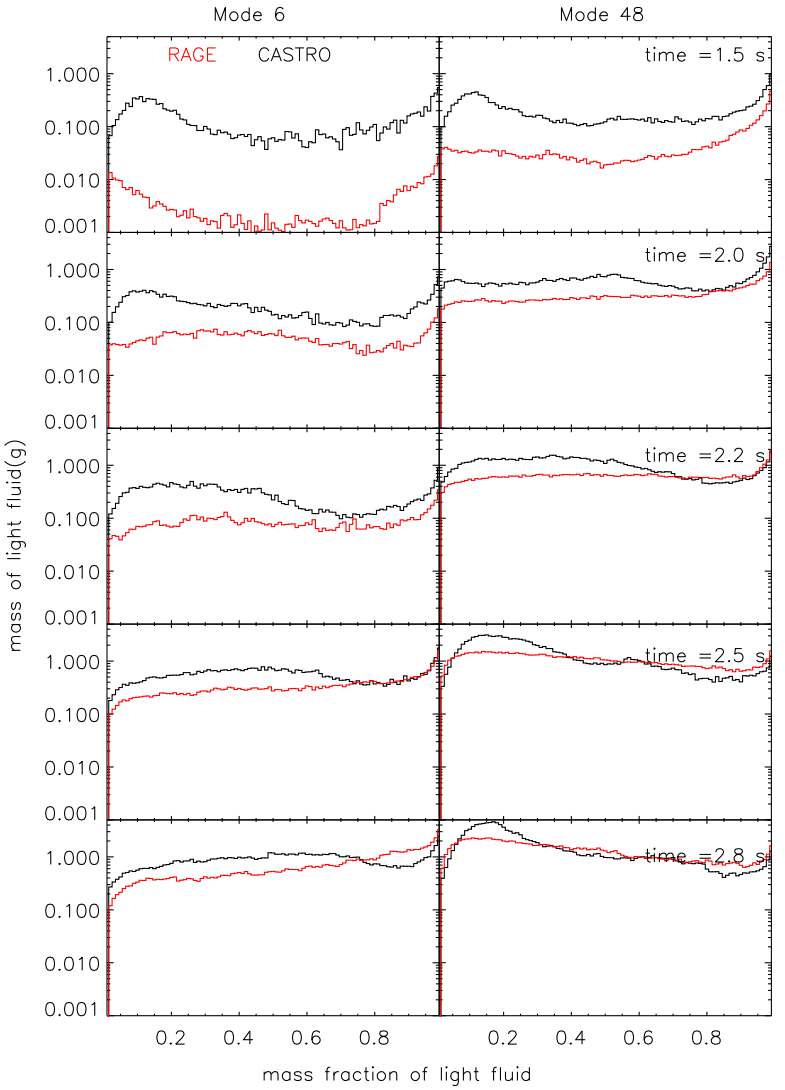

Fig. 16. Mass of light fluid binned by concentration for mode 6 (left) and mode 48 (right) simulations at times from the onset of mixing through the end of the calculation. CASTRO values are shown in red, while RAGE values are shown in black. RAGE shows much less mass in the mixed bins, those nearer the center of the figure, than does CASTRO at nearly all times during the simulation. Even at late times in simulations with a great deal of self-interaction among the fingers of the instability, CASTRO shows a great deal more light fluid mass at lower concentrations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The bubble and spike amplitudes agree between the simulations performed on Cartesian and spherical coordinate grids, in both cylindrical and spherical geometry. For cylindrical geometry, the small differences between Cartesian codes, in terms of bubble and spike amplitudes, are larger than the differences between Cartesian and polar-coordinate codes. For spherical geometry, the amplitude of perturbation growth is essentially the same between the three codes for both the low- and high-mode perturbations ($m = 6$ and $m = 48$). The presence or absence of a scheme to limit numerical diffusion, like the interface preserver employed by RAGE, impacts metrics of "mixedness" that become important later on in the growth of the instability, when instability growth is no longer completely characterized by bubble/spike amplitudes. Comparing the positions (relative to the 1D trajectory) of the average bubble and spike positions provides a good measure of overall growth rate of a fluid instability. This is the metric that would have the largest effect on supernova dynamics.

We have shown in this paper that for systems that are sufficiently perturbed, Cartesian meshes produce essentially the same results as cylindrical or spherical meshes for a cylindrical or spherical collapse problem, modulo small scale features arising from individual code differences. If the results are the same, are there other advantages to using a Cartesian mesh? For problems like this, one of the advantages is that the Courant time is uniform across the simulation. On a spherical mesh, the smaller cells at the center of the simulation give rise to Courant times that can be prohibitively small. In practice, this is avoided by cutting out the center of the simulation or restricting refinement there. Spherical coordinate codes also have a preferred center, by their very nature. Cartesian codes have no such preferred center, and of course no restrictive

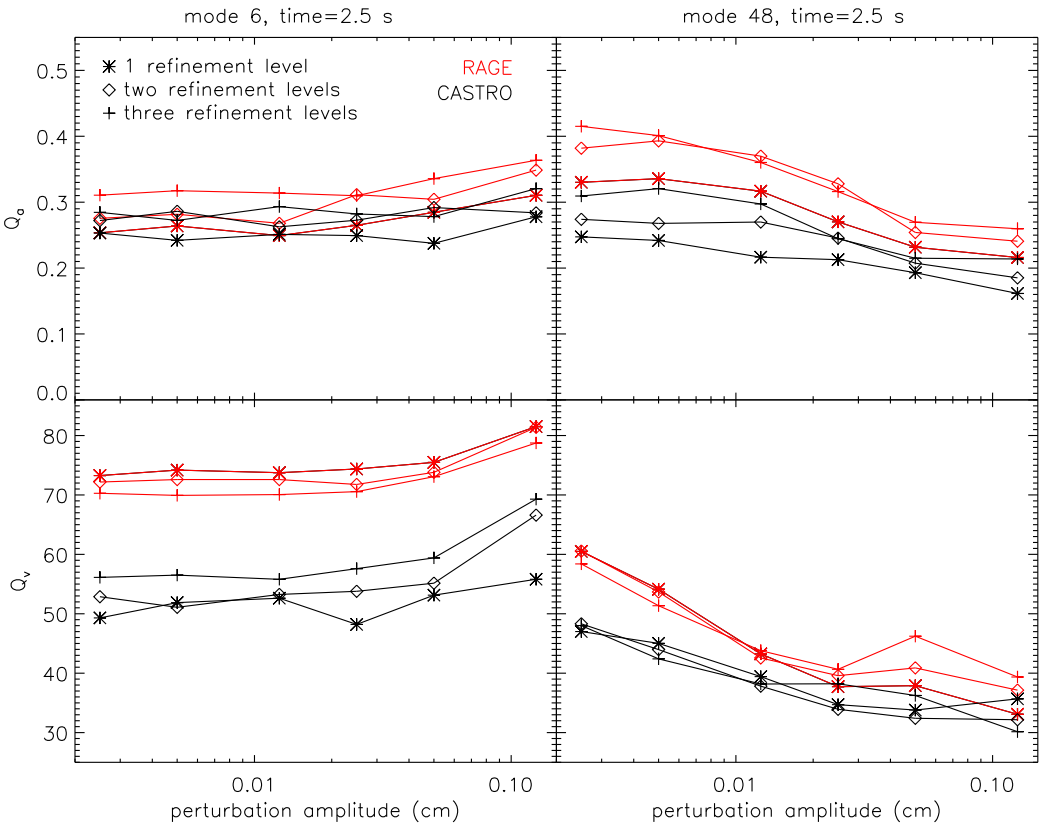

Fig. 17. Values for Q_a, (see Eq. (7)) and Q_v, (see Eq. (8)), and E_p for RAGE (in black) and CASTRO (in red). Q_v for the low-mode perturbation is most sensitive to numerical diffusion, and shows much higher values for RAGE, which explicitly limits numerical diffusion than CASTRO, which does not. This same value is closer between the codes for the high-mode perturbations, in which instability interaction helped further mix the fluid, but neither of these quantities will converge between codes as a function of increasing resolution. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Courant limit in the center of the simulation, as all cells are the same size throughout the domain at a particular level of resolution (for an AMR code) or a particular time (for a moving mesh code like PPM). Provided the resolution is sufficiently high, Cartesian codes are just as good, and in some respects better, than spherical-coordinate codes for following instability growth in a convergent geometry. While an aligned coordinate system is the only way to preserve an unperturbed interface in a multidimensional simulation, most real-world scenarios would experience some sort of perturbation. Indeed, it is the perturbations and their growth that makes this an interesting, multidimensional problem in the first place.

There are problems with taking the results of any 2D investigation of fluid instability growth too seriously. 2D geometries experience artificial drag forces that can slow the growth of instabilities [23,24]. The artificial drag forces will be different between 2D cylindrical and 2D axisymmetric calculations, as well. To that end, these simulations should not be taken as realistic representations of the growth rate of instabilities in an implosion geometry. 3D calculations are necessary to accurately determine the growth rate. The calculations presented herein serve to validate our proposed method of modeling these instabilities.

These simulations lay the groundwork for planned three-dimensional simulations of full-sphere implosions in Cartesian coordinates. By showing that for some amplitude of perturbation, results given by Cartesian and curvilinear coordinate grids are the same we have gained confidence in the accuracy of our 3D results. We have shown the minimum amplitude at which imposed perturbations swamp perturbations arising from the grid for convergent problems performed on a Cartesian grid. The minimum amplitude of perturbation for which convergent-geometry results obtained on Cartesian grids will be valid is expected to be significantly smaller than perturbations present in astrophysically relevant problems, such as supernova collapse. There the important perturbations arise from convective processes both prior to and during the explosion, and these are expected to be on the order of 10% of the wavelength of the perturbation [25,26].

The similarity of calculations performed on Cartesian meshes to those performed on polar meshes gives us confidence in extending these calculations to 3D. Issues that arose in 2D simulations—for example, the flow along axes, or a distortion in the reflected shock at the origin arising in part from flow along the y-axis—would not be present in 3D. The artificial drag forces present in 2D would also be absent.

Acknowledgements

This work was performed under the auspices of the United States Department of Energy (DOD) by Los Alamos National Security, LLC, at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396. The software used in this work was in part developed by the DOE NNSA-ASC OASCR Flash Center at the University of Chicago.

References

[1] M. Brouillette, The Richtmyer–Meshkov instability, Annu. Rev. Fluid Mech. 34 (2002) 445–468, http://dx.doi.org/10.1146/annurev.fluid.34.090101.162238.
[2] G. Dimonte, D.L. Youngs, A. Dimits, S. Weber, M. Marinak, S. Wunsch, C. Garasi, A. Robinson, M.J. Andrews, P. Ramaprabhu, A.C. Calder, B. Fryxell, J. Biello, L. Dursi, P. MacNeice, K. Olson, P. Ricker, R. Rosner, F. Timmes, H. Tufo, Y.-N. Young, M. Zingale, A comparative study of the turbulent Rayleigh–Taylor instability using high-resolution three-dimensional numerical simulations: the Alpha-Group collaboration, Phys. Fluids 16 (2004) 1668–1693, http://dx.doi.org/10.1063/1.1688328.
[3] Y. Zhou, B.A. Remington, H.F. Robey, A.W. Cook, S.G. Glendinning, A. Dimits, A.C. Buckingham, G.B. Zimmerman, E.W. Burke, T.A. Peyser, W. Cabot, D. Eliason, Progress in understanding turbulent mixing induced by Rayleigh–Taylor and Richtmyer–Meshkov instabilities, Phys. Plasmas 10 (2003) 1883–1896, http://dx.doi.org/10.1063/1.1560923.
[4] A.L. Velikovich, J.P. Dahlburg, A.J. Schmitt, J.H. Gardner, L. Phillips, F.L. Cochran, Y.K. Chong, G. Dimonte, N. Metzler, Richtmyer–Meshkov-like instabilities and early-time perturbation growth in laser targets and Z-pinch loads, Phys. Plasmas 7 (2000) 1662–1671, http://dx.doi.org/10.1063/1.873986.
[5] D.L. Youngs, R.J.R. Williams, Turbulent mixing in spherical implosions, Int. J. Numer. Methods Fluids 56 (2008) 1597–1603, http://dx.doi.org/10.1002/fld.1594.
[6] M. Gittings, R. Weaver, M. Clover, T. Betlach, N. Byrne, R. Coker, E. Dendy, R. Hueckstaedt, K. New, W.R. Oakes, D. Ranta, R. Stefan, The RAGE radiation-hydrodynamic code, Comput. Sci. Discov. 1 (1) (2008) 015005, http://dx.doi.org/10.1088/1749-4699/1/1/015005.
[7] A.S. Almgren, V.E. Beckner, J.B. Bell, M.S. Day, L.H. Howell, C.C. Joggerst, M.J. Lijewski, A. Nonaka, M. Singer, M. Zingale, CASTRO: a new compressible astrophysical solver. I. Hydrodynamics and self-gravity, Astrophys. J. 715 (2010) 1221–1238, http://dx.doi.org/10.1088/0004-637X/715/2/1221.
[8] A. Dubey, L.B. Reid, K. Weide, K. Antypas, M.K. Ganapathy, K. Riley, D. Sheeler, A. Siegal, Extensible component based architecture for FLASH, a massively parallel, multiphysics simulation code, arXiv e-prints, arXiv:0903.4875.
[9] A.C. Calder, B. Fryxell, T. Plewa, R. Rosner, L.J. Dursi, V.G. Weirs, T. Dupont, H.F. Robey, J.O. Kane, B.A. Remington, R.P. Drake, G. Dimonte, M. Zingale, F.X. Timmes, K. Olson, P. Ricker, P. MacNeice, H.M. Tufo, On validating an astrophysical simulation code, Astrophys. J. Suppl. Ser. 143 (2002) 201–229, http://dx.doi.org/10.1086/342267.
[10] B. Fryxell, K. Olson, P. Ricker, F.X. Timmes, M. Zingale, D.Q. Lamb, P. MacNeice, R. Rosner, J.W. Truran, H. Tufo, FLASH: an adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes, Astrophys. J. Suppl. Ser. 131 (2000) 273–334, http://dx.doi.org/10.1086/317361.
[11] P. Ramaprabhu, G. Dimonte, P. Woodward, C. Fryer, G. Rockefeller, K. Muthuraman, P.-H. Lin, J. Jayaraj, The late-time dynamics of the single-mode Rayleigh–Taylor instability, Phys. Fluids 24 (7) (2012) 074107, http://dx.doi.org/10.1063/1.4733396.
[12] A.R. Miles, B. Blue, M.J. Edwards, J.A. Greenough, J.F. Hansen, H.F. Robey, R.P. Drake, C. Kuranz, D.R. Leibrandt, Transition to turbulence and effect of initial conditions on three-dimensional compressible mixing in planar blast-wave-driven systems, Phys. Plasmas 12 (5) (2005) 056317, http://dx.doi.org/10.1063/1.1894765.
[13] P. Colella, M.D. Sekora, A limiter for PPM that preserves accuracy at smooth extrema, J. Comput. Phys. 227 (2008) 7069–7076, http://dx.doi.org/10.1016/j.jcp.2008.03.034.
[14] A. Harten, P. Lax, B. van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Rev. 25 (1) (1983) 35.
[15] H. Yang, An artificial compression method for ENO schemes: the slope modification method, J. Comput. Phys. 89 (1990) 125, http://dx.doi.org/10.1016/0021-9991(90)90120-P.
[16] W. Rider, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112–152, http://dx.doi.org/10.1006/jcph.1998.5906.
[17] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115–173, http://dx.doi.org/10.1016/0021-9991(84)90142-6.
[18] P. Colella, P.R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, J. Comput. Phys. 54 (1984) 174–201, http://dx.doi.org/10.1016/0021-9991(84)90143-8.
[19] P.R. Woodward, The PPM compressible gas dynamics scheme, in: F. Grinstein, L. Margolin, W. Rider (Eds.), Implicit Large Eddy Simulation: Computing Turbulent Fluid Dynamics, Cambridge University Press, 2006.
[20] B. van Leer, Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection, J. Comput. Phys. 23 (1977) 276, http://dx.doi.org/10.1016/0021-9991(77)90095-X.
[21] P.R. Woodward, Numerical methods for astrophysicists, in: K.H. Winkler, M.L. Norman (Eds.), Astrophysical Radiation Hydrodynamics, Reidel, 1986, pp. 245–326.
[22] P. Woodward, D. Porter, W. Dai, T. Fuchs, T. Nowatzki, M. Knox, G. Dimonte, F. Herwig, C. Fryer, The piecewise-parabolic Boltzmann advection scheme (PPB) applied to multifluid hydrodynamics, Tech. Rep. LA-UR-10-01823, Los Alamos National Lab., 2010.
[23] N.J. Hammer, H.-T. Janka, E. Müller, Three-dimensional simulations of mixing instabilities in supernova explosions, Astrophys. J. 714 (2010) 1371–1385, http://dx.doi.org/10.1088/0004-637X/714/2/1371, arXiv:0908.3474.
[24] C.C. Joggerst, A. Almgren, S.E. Woosley, Three-dimensional simulations of Rayleigh–Taylor mixing in core-collapse supernovae with CASTRO, Astrophys. J. 723 (2010) 353–363, http://dx.doi.org/10.1088/0004-637X/723/1/353, arXiv:1009.3336.
[25] C.A. Meakin, D. Arnett, Anelastic and compressible simulations of stellar oxygen burning, Astrophys. J. 665 (2007) 690–697, http://dx.doi.org/10.1086/519372, arXiv:astro-ph/0611317.
[26] J. Nordhaus, A. Burrows, A. Almgren, J. Bell, Dimension as a key to the neutrino mechanism of core-collapse supernova explosions, Astrophys. J. 720 (2010) 694–703, http://dx.doi.org/10.1088/0004-637X/720/1/694, arXiv:1006.3792.